

МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Физтех-школа физики и исследований им. Ландау

Машинное обучение. Конспект

Автор:

Яренков Александр Владимирович

Долгопрудный
15 июля 2024 г.

1 Введение

Определим основные понятия и покажем их на примере из таблицы под этим предложением.

Name	Age	Statistics (mark)	Python (mark)	Eye color	Native language	Target (mark)	Predicted (mark)
John	22	5	4	Brown	English	5	1
Aahna	17	4	5	Brown	Hindi	4	5
Emily	25	5	5	Blue	Chinese	5	2
Michael	27	3	4	Green	French	5	4
Some student	23	3	3	NA	Esperanto	2	3

- Датасет/выборка/набор данных - данные, которые нам доступны
Вся таблица, кроме последней колонки.
- Объект/наблюдение/datum/data point - элемент выборки
Обычно(всегда ли?) мы предполагаем их независимыми и одинаково распределёнными (н.о.р.).
Строка в таблице
- Признак/feature - характеристика объекта.
Все колонки признаков образуют матрицу признаков/design matrix (X) (её столбцы x_i - вектора, имеющие все признаки объекта).
Первые 6 колонок в таблице
- Целевая (зависимая) переменная/target (y) - переменная, значение которой нас интересует
В задаче бинарной классификации (target принимает значения из множества мощностью 2) имеет специфическое название - label. Задача регрессии ($y \in \mathbb{R}$)
Жёлтый столбец в таблице
- Модель - метод обучения. Формально, действующее на признаковом пространстве отображение ($f(X)$)
- Предсказание/prediction (\hat{y}) - результат работы модели
Зелёная колонка в таблице
- Функция потерь/Loss (L) - критерий качества модели. Её значение мы пытаемся уменьшать, чтобы модель была качественнее.
Примеры функции потерь:

Mean Squared Error/дисперсия: $MSE(y, \hat{y}) = \|y - \hat{y}\|_2 = \frac{1}{N-1} \sum_{i=1}^N (y_i - \hat{y}_i)^2$

Mean Absolute Error: $MAE(y, \hat{y}) = \|y - \hat{y}\|_1 = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$

Здесь N — количество наблюдений

- Параметры - Переменные, значение которых модель выбирает сама
Например, взвешивание данных
- Гиперпараметры - параметры, которые мы фиксируем до начала работы с данными
Например, глубина решающего дерева

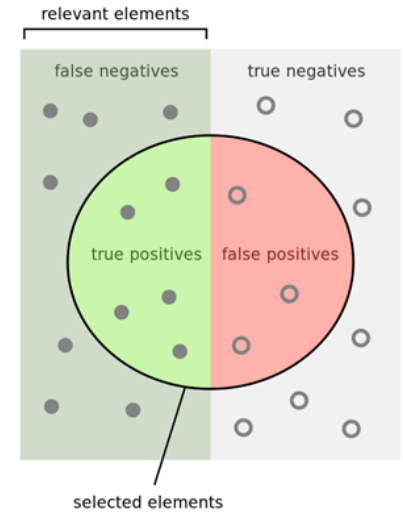
1.1 Формальная задача обучения с учителем

Обучение с учителем - обучение с наличием целевой переменной.

Поставим формально задачу обучения с учителем: $\{x_i, y_i\}_{i=1}^N$ — тренировочный датасет (training set), $f(X)$ — модель f , которой на вход подаём матрицу признаков X и получаем предсказание $f(X) = \hat{y}$, $L(X, y, f)$ — Loss function, that should be minimized

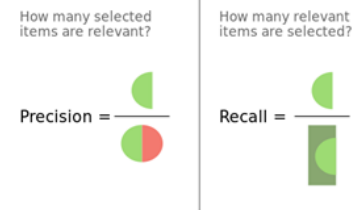
1.2 Метрики

		True condition	
		Condition positive	Condition negative
Predicted condition	Predicted condition positive	True positive TP	False positive, FP Type I error
	Predicted condition negative	False negative, FN Type II error	True negative TN



Precision - доля верно классифицированных объектов (TP) ко всем классифицированным объектам (TP + FP). Она же точность

Recall - доля верно классифицированных объектов (TP) ко всем объектам этого класса (TP + FN). Т.е. мы узнаём сколько значений из класса мы НЕ потеряли.



Есть задачи в которых важнее либо precision, либо recall. Например, заболевшие люди приходят в больницу. Если им назначить лечение, не факт что им станет лучше. Но если им не назначить лечение, то им может стать сильно хуже. Поэтому нам не так важно улучшить состояние каждого (повысить precision - в данном случае доля заболевших людей, у которых улучшилось состояние), как чтобы людям не становилось ещё хуже (т.е. нужно повысить recall - в данном случае доля заболевших людей, у которых не ухудшилось состояние)

Поэтому введём понятие F_β -score. Это метрика, учитывающая важность либо precision, либо recall.

$$F_\beta = (1 + \beta^2) \frac{Precision \cdot Recall}{\beta^2 Precision + Recall}$$

Наиболее частые частные случаи:

$$F_\beta|_{\beta=0} = F_0 = 2 \cdot Precision$$

$$F_\beta|_{\beta=+\infty} = F_\infty = Recall$$

$$F_1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

2 Наивный байесовский классификатор

$\{x_i, y_i\}_{i=1}^N$ - training set. $x_i \in \mathbb{R}^p$ - вектора, $y_i \in \{C_i\}_{i=1}^k$ (Задача классификации на k классов).

А также наивное предположение, которое делает байесовский классификатор наивным: все признаки независимые.

Согласно теореме Байеса,

$$P(y_i = C_j | x_i) = \frac{P(x_i | y_i = C_j) \cdot P(y_i = C_j)}{P(x_i)} \quad (1)$$

Так как все признаки считаем независимыми, то происходит следующая факторизация:

$$P(x_i | y_i = C_j) = \prod_{l=1}^p P(x_i^l | y_i = C_j)$$

Таким образом вероятность $P(x_i | y_i = C_j)$ мы поняли как считать. Вероятность $P(y_i = C_j)$ мы сможем посчитать по частоте встречаемости экземпляров класса в тренировочной выборке.

Так как у нас стоит задача классификации, то ответом к ней будет класс, к которому относится объект: $C = \arg \max_j P(y_i = C_j | x_i)$ - аргумент, при котором вероятность максимальна. Заметим, что максимизируем мы по различным классам, а знаменатель выражения (1) не зависит от j , поэтому для задачи максимизации он нам не нужен и мы его просто выбросим из рассмотрения. При желании его посчитать, можно воспользоваться формулой полной вероятности.

3 Градиентная оптимизация

Пусть модель зависит от параметров w : $f_w(X)$. Наша задача в повышении качества нейронной сети. Математически это качество выражается в виде функции качества, или обратной к ней - функции потерь. Т.о. наша задача состоит в минимизации функции потерь.

Для этого подберём параметры модели, при которых функция потерь будет наименьшей (**это и есть оптимизация**). Так как градиент - это направление наискорейшего роста, то будет подбирать параметры модели следующим образом:

$$w_{n+1} = w_n - \alpha \nabla_w L(y, f_w(X))$$

где (w_n) - последовательность параметров модели (при $n \rightarrow \infty$ последовательность должна сходиться к локальному минимуму), w_0 - начальные значения параметров, α - шаг оптимизации/learning rate - некоторый коэффициент, который может меняться на каждом шагу.

3.1 Условная оптимизация

Условная оптимизация - это оптимизация параметров, на которые наложены ограничения.

Например, в нашей задаче есть n классов и в каждом из них какое-то количество объектов. Вероятность посмотреть на элемент $i_{\text{того}}$ класса равна p_i . Эти вероятности и будем считать параметрами. На них есть понятное ограничение: $\sum_{i=1}^n p_i = 1$. Это простой случай, т.к. это ограничение со знаком равенства, а значит будем применять метод множителей Лагранжа (3й семестр мат. анализа) для нахождения условных экстремумов.

В качестве основы для функции Лагранжа выберем энтропию (*почему?*) $S = - \sum_k p_k \ln p_k$

Тогда функция Лагранжа $\mathcal{L} = - \sum_k p_k \ln p_k + \lambda(1 - \sum_k p_k)$

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial p_i} = -\ln p_i - 1 + \lambda = 0 \\ 1 - \sum_k p_k = 0 \end{cases}$$

$$\begin{cases} \ln p_i = \lambda - 1 \Rightarrow p_i = \exp(\lambda - 1) \\ 1 - \sum_k p_k = 0 \Rightarrow 1 - \sum_k \exp(\lambda - 1) = 0 \Rightarrow 1 - k \exp(\lambda - 1) = 0 \Rightarrow \lambda = 1 - \ln k \end{cases}$$

Таким образом $\ln p_i = -\ln k$ или $p_i = \frac{1}{k}$. Мы нашли кандидатов на условный экстремум и дальше делаем с ними тоже самое, что и для градиентной оптимизации.

4 Методы регрессионного анализа

4.1 Линейные модели и линейная регрессия

Модель $f(X)$ называется линейной, если она является линейной комбинацией признаков и какого-то числа.

Поставим формально задачу линейной регрессии: $\{x_i, y_i\}_{i=1}^N$ — training set, $x_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}$

$$f(X) = \omega_0 + \sum_{k=1}^p \omega_k x_k$$

Свободный член w_0 называют **bias term**, т.е. параметр смещения. Потому что в линейной функции свободный член отвечает за смещение прямой относительно начала координат.

Эта запись выглядит довольно громоздко. Её можно упростить с помощью скалярного произведения признаков на веса. Но тогда останется bias term w_0 . Поэтому, чтобы писать только скалярное произведение введём следующие обозначения.

Обозначим $\tilde{X} = (1, x_1, x_2, \dots, x_p)^T$, а $W = (w_0, w_1, w_2, \dots, w_p)^T$ — вектор весов. Тогда $f(X) = \tilde{X}^T W$

Для решения задачи нам нужно найти такие веса W , которые минимизируют функцию потерь. Эти искомые веса $\hat{W} = \arg \min_w L$. Тогда решением задачи/предсказанием модели будет $\hat{Y} = XW$

Обычно задачи машинного обучения не имеют аналитического решения, но если мы выберем функцию потерь $L = \|Y - XW\|_2^2$ и будем её минимизировать, то по сути мы будем применять МНК.

$$L = (Y - XW)^T (Y - XW)$$

$$\nabla_w L = \nabla_w (Y^T Y + W^T X^T X W - Y^T X W - W^T X^T Y) = 0 + X^T X W \cdot 2 - X^T Y \cdot 2 = 0 \Rightarrow$$

$W = (X^T X)^{-1} X^T Y$. Это решение существует, только если матрица $X^T X$ обратима, т.е. её детерминант не равен 0, т.е. все признаки линейно независимы. Есть даже **Теорема Гаусса-Маркова**, которая утверждает что если $Y = XW + \varepsilon$, где $\varepsilon = (\forall i \mathbb{E} \varepsilon_i = 0)$, дисперсии конечны, а $\forall i, j \text{ cov}(\varepsilon_i, \varepsilon_j) = 0$, то решение найденное по МНК — оптимальное несмещённое решение.

Но что делать в случае когда признаки зависимы?

4.2 Регуляризация

Т.к. признаков стало меньше, то для сохранения единственности решения необходимо ввести дополнительные ограничения. Такое введение ограничений называется регуляризацией.

Например, можно добавить диагональную матрицу для получения линейной независимости и самый простой способ — добавить единичную матрицу. $W = (X^T X + \lambda E)^{-1} X^T Y$. На самом деле

такой вектор весов это будет решение с помощью МНК для следующей функции потерь: $L = \|Y - XW\|_2^2 + \lambda^2 \|W\|_2^2$. Это есть L_2 регуляризация. Частный случай L_p регуляризации.

Вообще L_2 регуляризация пытается найти минимум используя все признаки, т.к. имеет меняющийся знак в нуле линейный градиент, а L_1 регуляризация пытается занулить наибольшее количество весов, т.к. имеет меняющийся знак в нуле константный градиент.

4.3 Задача классификации

Введём понятие правдоподобия. Пусть наша модель работает с параметрами θ . Тогда функция правдоподобия - это вероятность $P(X, Y|\theta)$ пронаблюдать такие X и Y при параметрах θ . Логично, что функцию правдоподобия нужно максимизировать. Если X и Y одинаково распределённые $\forall i$ x_i и y_i независимые, то $P(X, Y|\theta) = \prod_i P(x_i, y_i|\theta)$. Далее мы можем воспользоваться монотонной функцией логарифм, т.е. максимизация логарифма величины это та же задача, что и максимизация самой величины. Тогда мы будем рассматривать $\sum_i \ln P(x_i, y_i|\theta) \rightarrow \max_{\theta}$

4.4 Логистическая регрессия

Регрессия называется логистической, т.к. есть "0" и "1". Соответственно имеем задачу бинарной классификации. Пусть $p \in [0, 1]$ - вероятность принадлежности к одному из классов. Но мы пока умеем решать только задачу регрессии, и предсказание является элементом \mathbb{R} . Поэтому нам нужно провести какие-то преобразования с вероятностью p , такие что сохранится монотонность вероятности и новое выражение будет принимать значение из \mathbb{R} .

Один из способов следующий:

$$p \rightarrow \frac{p}{1-p} \in [0, \infty) \rightarrow \ln \frac{p}{1-p} \in \mathbb{R}$$

И уже это выражение можно приравнять к предсказанию линейной регрессии $f(X)$ (см. 4.1). Выражая отсюда p , получим $p = \frac{1}{1 + e^{-f(X)}}$. Полученная функция называется **сигмоида**. $\sigma(f(X))$

Для задачи бинарной классификации определим понятие *Margin* (сокр. M). Введём каждому из двух классов метку: положительную 1 или отрицательную -1 . Тогда $M = \text{метка класса} \cdot f(X)$.

Итак, вероятность принадлежности к одному из классов, согласно выкладкам выше, есть $\sigma(f(X)) = \sigma(M)$. Тогда вероятность принадлежности к другому классу с *отрицательной меткой* есть $1 - \sigma(f(X)) = \sigma(-f(X)) = \sigma(M)$.

Таким образом, для максимизации функции правдоподобия, нам нужно максимизировать $\sum_i \ln P(x_i, y_i|\theta) = - \sum_i \ln (1 - \exp(-M_i))$ или $\sum_i \ln (1 - \exp(-M_i)) \rightarrow \min_w$. Последнюю называют *логистической функцией потерь*, которую минимизируют.

4.5 Многоклассовая классификация

Чаще всего пользуются следующим методом: с помощью логистической регрессии относят данные к каждому классу по отдельности. Т.е. сколько классов, столько логистических регрессий и нужно будет построить. А потом логично с помощью биссектрис (равноудаление от линий, полученных в результате регрессии) углов добавить оставшиеся данные.

